

CS 610 Semester 2022–2023-I: Assignment 4

17th October 2022

Due Your assignment is due by Oct 28, 2022, 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not plagiarize or turn in solutions from other sources. You will be PENALIZED if caught.

Submission

- Submission will be through mooKIT.
- Name each source file as “<roll-no-probno>.cpp” (e.g., “22111090-prob2.cpp”).
- Submit a PDF file with name “<roll-no>.pdf”. Describe your solutions to the problems in the PDF, and explain your implementations. Describe any challenges (if any) that you may have faced.
- Provide exact compilation instructions. The TBB examples compile with TBB (versions 2017 U7-8 and 2020.1.2) available on Ubuntu 18.04 and 20.04 LTS releases.
- We will provide template code for a few problems. Modify the template code as instructed and perform evaluations.
- We encourage you to use the L^AT_EX typesetting system for generating the PDF file.
- You will get up to TWO LATE days to submit your assignment, with a 25% penalty for each day.

Evaluation

- Your solutions should only use concepts that have been discussed in class.
- Show your computations where feasible and justify briefly in the PDF.
- Write your code such that the EXACT output format is respected.
- We will evaluate the implementations with our OWN inputs and test cases (where applicable), so remember to test thoroughly.
- We may deduct marks if you disregard the listed rules.

Problem 1

[10+20+20 marks]

- (i) Implement Fibonacci computation using explicit OpenMP tasks (for e.g., `task` or `sections`). You can use a cutoff to fallback to serial code for better performance.
- (ii) Implement Fibonacci with the blocking style parallelism in Intel TBB.
- (iii) Implement Fibonacci with continuation passing style parallelism in Intel TBB.

Compare performance of the three implementations with the serial code and explain your observations.

Problem 2

[30 marks]

Parallelize the quick sort implementation with OpenMP. Compare and report the performance of the serial and parallel versions.

Problem 3

[30+10 marks]

Use Intel TBB parallel algorithms to compute the value of π in parallel. Compare and report the performance of the serial version, an OpenMP version, and the parallel TBB version.

Pick an OpenMP version that works best for you, it need not be one we discussed in class. Try to minimize the impact of false sharing for your parallel implementations.

Problem 4

[30 marks]

Implement a parallel version of “find max” for a given array of numbers using Intel TBB parallel algorithms. Compare and report the performance of the serial version and the parallel version. Assume the max size of the input array to be limited to 2^{26} .

Report the index of the first occurrence of the max value in case there are duplicates.